WATCHDOG TIMER FOR MICROCONTROLLER

Field of the Invention

[0001] The present invention relates to monitoring timers, and in particular, to monitoring timers associated with microcontrollers. These monitoring timers are also called watchdog timers.

Background of the Invention

[0002] Microcontrollers associated with watchdog timers are known in the prior art. External interference or unexpected logic conditions can sometimes cause failure of an application or software used by the microcontroller. The closer controllers are located to strong magnetic and electrical fields, the more they are subject to such failures. The application or the software then fails to operate in accordance with its normal logic sequence.

[0003] Watchdog timers conventionally have a register with several bits which store a remaining time period before a reset. The bits are decremented at regular intervals by a timer, in the form of a countdown. When the microcontroller operates normally, the micro-controller writes a non-zero time period at regular intervals in the register. In most cases of abnormal microcontroller operation, the application or software runs in a loop and no longer writes any new

time value in the register of the watchdog timer. The countdown therefore continues until it times out.

[0004] On timeout, the watchdog timer resets the microcontroller. For this purpose, one of the register bits is logically connected to a microcontroller reset pin. When this bit reaches a low status, i.e., on expiration of the time limit, the microcontroller is reset. The application or software used on the microcontroller can then be restarted under good conditions. A logic circuit can also be inserted between the register bit intended for resetting and the reset pin of the microcontroller. The logic circuit can be an ET or NON-ET type and have a second input connected to a logic activation input. The functioning of the watchdog timer can therefore be activated or disabled selectively via this circuit.

[0005] This type of watchdog timer has disadvantages. The software or application of the microcontroller may unduly write new time periods at reduced time intervals in the register of the watchdog timer. This may be the case, in particular, if the application or software of the microcontroller runs in an endless loop on a write routine in the register of the watchdog timer. Therefore, even in the event of failure of the application or program, the register does not undergo countdown to timeout and the application or software is therefore not reset.

[0006] Also, U.S. Patent No. 4,705,970 describes a watchdog timer for a microprocessor. The timer has two counters. The counters are respectively associated with a minimum time period and a maximum time period separating two successive timer refresh commands. These two counters are fully independent. This timer has the disadvantage of two close refresh commands causing

resetting of the microprocessor, whereas they do not necessarily correspond to a software loop.

Summary of the Invention

[0007] In view of the foregoing background, an object of the present invention is to provide a microcontroller watchdog timer that addresses the above stated disadvantages.

[0008] This and other objects, advantages and feature in accordance with the present invention are provided by a microcontroller watchdog timer comprising a reset counter having a refresh input and a reset output. The microcontroller watchdog timer starts a reset countdown on receipt of a refresh command on its refresh input, and applies a reset command to the reset output on expiration of the reset countdown.

[0009] The microcontroller watchdog timer in accordance with the present invention, which conforms to the generic definition given in the background section, is essentially characterized in that it applies a method for optimizing its functioning, as described below.

[0010] According to a first embodiment, the watchdog timer comprises a reset counter comprising a refresh input and a reset output. The watchdog timer starts a reset countdown on receipt of a refresh command on its refresh input, and applies a reset command to the reset output on timeout of the reset countdown. A refresh counter is connected to the refresh input of the reset counter, and has a refresh input and starts a refresh countdown on receipt of a refresh command on its refresh input. A refresh command is only applied to the refresh input of the reset counter on timeout of the refresh countdown time.

[0011] According to a second embodiment, the watchdog timer comprises a reset counter having a refresh input and a reset output, and starts a reset countdown on receipt of a refresh command on its refresh input, and applies a reset command to the reset output on timeout of the reset countdown. A refresh counter is connected to the refresh input of the reset counter, and has a refresh input, and starts a refresh countdown on receipt of a refresh command on its refresh input. A refresh command is only applied to the refresh input of the reset counter if the refresh countdown has expired at the time of receipt of the refresh command.

[0012] The reset counter may comprise a register. The register may comprise a plurality of bits including a high-order bit connected to the reset output.

[0013] The watchdog timer may also comprise an activation input. A logic circuit having a first input may be connected to the activation input, and a second input of the logic circuit may be connected to the reset output of the reset counter. The register may also include an activation bit connected to the activation input.

[0014] Provision may also be made for the watchdog timer to have a timer input, and a frequency divider. The frequency divider may have an input connected to the timer input, and an output may be connected to the bits of the reset counter for decrementing the reset counter when a signal is applied thereto by the frequency divider.

[0015] The duration of the refresh countdown may be shorter than the time separating two decrements of the reset counter. The refresh counter may comprise a plurality of bits and may also be connected to the

timer input. The refresh counter is decremented when a clock signal is applied thereto. The duration of the reset countdown may be programmable.

[0016] The invention also concerns a method for optimizing the functioning of a microcontroller watchdog timer having normal time periods separating refresh commands of the watchdog timer lying within a range having a minimum time period and a maximum time period. The method comprises receiving refresh commands by the watchdog timer, and generating a reset command for the microcontroller by the watchdog timer when the time interval separating successively received refresh commands is not within the range.

[0017] Generating the reset command may comprise launching or starting a reset countdown and generating the reset command on expiration of the reset countdown. The refresh countdown is restarted for each refresh command received. If a refresh command is received while the refresh countdown has not timed out, a refresh command does not restart the reset countdown.

[0018] The method may comprise starting a reset countdown, on receipt of a refresh command by the watchdog timer, and start of the refresh countdown. The method may further comprise restarting of the reset countdown on expiration of the refresh countdown, and generating a reset command on expiration of the reset countdown. The duration of the reset countdown is equal to the maximum time period of the range, with the duration of the refresh countdown being equal to the minimum time period of the range.

[0019] On receipt of a refresh command by the watchdog timer, if the refresh countdown has timed out, the reset countdown is started, and if the refresh

countdown has not timed out, then the reset countdown continues and the refresh countdown is restarted.

[0020] A reset command may be generated on expiration of the reset countdown. The duration of the reset countdown may be equal to the maximum time period of the range, with the duration of the refresh countdown being equal to the minimum time period of the range.

[0021] If the refresh countdown has timed out, the start of the reset countdown and the start of the refresh countdown may be simultaneous on receipt of a refresh command. The reset countdown may be synchronized on a frequency-divided clock signal. Likewise, the refresh countdown may be synchronized on a clock signal.

[0022] The method may also comprise a programming step for programming the duration of the reset countdown. The refresh command received by the refresh input of the refresh counter may define the duration of the reset countdown.

[0023] The refresh command may comprise a word made up of a plurality of bits written in the reset counter during a starting step for defining the duration of the reset countdown. The reset counter may comprise a counter including a plurality of bits, and the reset command may be generated by transition of the high-order bit of the reset counter to a low level.

Brief Description of the Drawings

[0024] The invention will be better understood on reading the following description and examining the accompanying figures.

[0025] Figure 1 is a schematic illustration of a watchdog timer according to the present invention associated with a microcontroller;

[0026] Figures 2 to 4 are timing diagrams illustrating the functioning of a first example of a watchdog timer for different cases of microcontroller operation; and

[0027] Figures 5 to 7 are timing diagrams illustrating the functioning of a second example of a watchdog timer for different cases of microcontroller operation.

Detailed Description of the Preferred Embodiments

[0028] The invention therefore puts forward a watchdog timer generating a reset signal for a microcontroller when the time interval separating successive refresh commands is shorter than a minimum time period for normal operation. The invention particularly concerns a watchdog timer in which a refresh counter locks refresh commands intended to restart a reset counter of the microcontroller. The refresh commands are given by the microcontroller and their transmission to the reset counter is locked during the countdown of the refresh counter. The refresh commands, received by the refresh counter before its time is out, reset this counter. Therefore, if the microcontroller sends refresh commands in a loop, the countdown of the reset counter is not restarted and can reset the microcontroller when it expires. A refresh loop of the microcontroller therefore does not inhibit resetting of the microcontroller.

[0029] The invention can be applied in particular using two embodiments described below. According to a

first embodiment, the refresh commands are inhibited during the refresh counter countdown and are only transmitted to the reset counter when it expires. According to a second embodiment that is even more advantageous, the refresh commands are not transmitted when they arrive during the refresh countdown. In both embodiments, the received refresh commands restart the countdown of the refresh counter.

[0030] Figure 1 is a schematic illustration of the structure of a watchdog timer 1 according to the invention, and an associated microcontroller 2. The refresh clock is delimited in Figure 1 by the dashed line. The watchdog timer 1 has a reset command output pin 8 connected to a reset pin RST on the microcontroller 2. This pin 8 is therefore designed to apply, in a known manner, a reset signal to an appropriate pin of the microcontroller 2. The watchdog timer 1 also has a refresh command input pin 9 on which a pin Raf of the microcontroller 2 can apply a refresh command.

[0031] The watchdog timer 1 also has a reset counter 3 as readily understood by those skilled in the art. This counter 3 has a reset command output 12 having a logical link with the reset command output pin 8. The reset command output 12 can therefore either directly apply the reset signal to pin 8, or be connected to pin 8 via a logic port as will be detailed below. The reset counter 3 also has a refresh command input 11.

[0032] The reset counter 3 is designed to apply a reset command to the reset output 12 on expiration of its countdown. The reset counter 3 is also designed to start a new countdown when a refresh command is applied to the refresh command input 11.

[0033] A refresh counter 6 is connected between the refresh command input terminal 9 and the refresh command input 11. The refresh counter 6 is designed to launch or start a new refresh countdown, i.e., its own countdown, on receipt of a refresh command on the refresh command input terminal 9.

[0034] According to the first embodiment, this refresh command is only transmitted to the refresh command input 11 of the reset counter 3 if no refresh command is received before the timeout of the refresh countdown.

[0035] According to the second embodiment, such a refresh command is not transmitted to the refresh command input 11 of the reset counter 3 if it is received during the countdown of the refresh counter 6.

[0036] Therefore, assuming that the microcontroller 2 runs an endless loop sending close refresh commands, the reset countdown 3 is nonetheless not restarted. The reset counter 3 then continues its countdown while the refresh counter 6 is restarted on each refresh command, i.e., indefinitely since the microcontroller 2 sends these commands in a loop. When the reset counter 3 reaches timeout, it emits a reset command to the microcontroller 2 to reset the application or software which has gone into a loop.

[0037] The watchdog timer 1 is therefore designed only to leave the microcontroller 2 operating continuously if the microcontroller generates refresh command signals separated by normal time intervals lying within a given range.

[0038] Figures 2 to 4 show timing diagrams of the functioning of the watchdog timer, according to the first embodiment. Figure 2 is a timing diagram of a normal operation. The maximum desired interval

separating refresh commands is equal to the duration of the reset countdown. If no refresh command is given during this time interval, the countdown of the reset counter times out as illustrated in Figure 3. The duration of the refresh countdown is also equal to the minimum time period of the range as illustrated in Figure 4.

[0039] Figure 2 illustrates the functioning of the watchdog timer 1 during normal operation of the microcontroller 2. The microcontroller 2 is able to send refresh commands spaced out by a time interval lying within the desired range. The first timing diagram shows the status of the refresh counter 6 while the second timing diagram shows the status of the reset counter 3. The countdown of the reset counter 3 is started at time Init. The countdown of the refresh counter 6 is started when it receives a refresh command C-Raf. On expiration (time denoted Exp-TRaf or Raf) of a TRaf time interval, the countdown of the reset counter 3 has not timed out, and this countdown is restarted. The reset counter 3 therefore does not generate a reset command. Consequently, a maximum time period is defined separating refresh commands, sent by the microcontroller 2, by fixing the duration of the reset countdown.

[0040] Figure 3 illustrates the functioning of the watchdog timer 1 during a microcontroller loop which does not generate a refresh command. The microcontroller 2 operates correctly up until time Def. It then no longer sends out a refresh command. The reset countdown subsequently continues its countdown until it times out. The watchdog timer 1 then generates, in a known manner, a reset command at time Reinit, as shown in the third timing diagram of Figure 3. The refresh counter 6 therefore has no influence on

the generation of a reset command in this particular case.

[0041] Figure 4 shows the functioning of the watchdog timer 1 during a microcontroller loop generating close refresh commands initiated at time Def. When the time interval separating successive refresh commands of the microcontroller 2 is shorter than the duration of the refresh countdown TRaf, the refresh countdown is restarted at each new refresh command. The countdown of the reset counter 3 is then not restarted. When this counter times out at time Reinit, the watchdog timer 1 generates a reset command, as shown in the third timing diagram of Figure 4. The duration of the refresh countdown TRaf therefore defines the minimum time period which must separate refresh commands given in repeated succession.

[0042] The watchdog timer 1 using this first embodiment advantageously comprises a memory component for the last refresh command received on input 9 so that if necessary it can transmit the same to the reset counter 3 on timeout of the refresh countdown.

[0043] Figures 5 to 7 are timing diagrams of watchdog timer operations according to the second embodiment. Figure 5 is a timing diagram of normal operation. The maximum desired period separating refresh commands is also equal to the duration of the reset countdown. If no refresh command is given during this period, the countdown of the reset counter 3 times out as illustrated in Figure 6. The duration of the refresh countdown is also equal to the minimum time period of this range, as illustrated in Figure 7.

[0044] Figure 5 therefore illustrates the functioning of the watchdog timer 1 according to the second embodiment during normal operation of the

microcontroller 2. The microcontroller 2 is able to send refresh commands spaced out by a time interval lying within the desired range. The countdown of the refresh counter 6 is started at the time corresponding to receipt of an initial refresh command C-Raf. At this time, the countdown of the refresh counter 6 is still in a timeout status. The refresh command is then transmitted to the reset counter 3. The countdown of the reset counter 3 is therefore initially restarted at the same time as the refresh countdown. The following refresh command is received after expiration of the countdown of the refresh counter 6. This command therefore also restarts the reset counter 3 at time Raf, before the timeout of the reset countdown then in progress. The reset counter 3 therefore does not generate a reset command. As in the previous embodiment, a maximum time period is defined separating refresh commands, sent by the microcontroller 2, by setting the duration of the reset countdown.

[0045] Figure 6 illustrates the functioning of the watchdog timer 1 during a microcontroller loop which does not generate a refresh command. The refresh counter 6 and the reset counter 3 are started simultaneously on receipt of a command C-Raf. The microcontroller 2 operates correctly up until time Def, then no longer emits a refresh command. The reset counter 3 then continues its countdown until it times out. The watchdog timer 1 then, in a known manner, generates a reset command at time Reinit, as shown in the third timing diagram in Figure 6. In this embodiment, the refresh counter 6 therefore does not have any influence either on the generation of a reset command for this type of loop.

[0046] Figure 7 illustrates the functioning of the watchdog timer 1 during a microcontroller loop

generating close refresh commands, initiated at time Def. Each C-Raf command restarts the countdown of the refresh counter 6 as described previously. Each subsequent C-Raf command is therefore received before the timeout of the refresh counter 6 countdown. These commands are therefore not transmitted to the reset counter 3. The countdown of the reset counter 3 is therefore not restarted and times out at time Reinit. A reset command is then generated as shown in the third timing diagram in Figure 7. The duration of the refresh countdown Traf therefore defines the minimum time period which is to separate refresh commands given in repeated succession.

[0047] It may be provided that the restart of the reset countdown is made via a microcontroller write in the reset counter 3. In particular, provision may be made for this write to be inhibited during the refresh countdown. This second embodiment is advantageous since it does not require memorizing or time shifting a refresh command received on the input 9 and transmitting it to the reset counter 3.

[0048] In the variation shown in Figure 1, the reset counter comprises several bits B0 to B6. The high-order bit B6 of the reset counter 3 is connected to the reset output 12. Therefore, switching of the high-order bit B6 corresponds to expiration of the countdown and generates a reset signal given to the microcontroller 2. In the example shown, the reset counter 3 has 64 possible values during the countdown, defined by values B0 to B5.

[0049] The counter may be designed, for example, so that transition to low status of bit B6, complemented at the input of a logic device 7, applies a reset signal lasting on the order of 500 nanoseconds to a

reset pin of the microcontroller 2. The microcontroller 2 is adequately programmed so as to send refresh commands at time intervals such as defined previously.

[0050] Use of a logic device 7 makes it possible to couple the reset output 12 of the reset counter 3 with an activation input 13. With the activation input 13 it is therefore possible to selectively activate or disable the generation of reset commands on the output pin 8. In the example in Figure 1, the activation input fixes the status of an Act bit in a register in which the reset counter 3 is also included.

[0051] The watchdog timer 1 may also have a timer input 14 connected to the input of a frequency divider 5. The output 10 of the frequency divider 5 is connected to the reset counter 3. The frequency-divided signal is then used to decrement the reset counter 3. The clock signal is for example provided by an external oscillator 4, common to several electronic circuits. Provision may also be made for integration of the oscillator 4 in the watchdog timer 1 or in the microcontroller 2. The integrated oscillator 4 is then independent from the operating uncertainties of a possible external oscillator.

[0052] Provision may also be made for use of a refresh counter 6 having several bits, connected to the timer input, to decrement the countdown on each received clock signal. The refresh counter 6 has a connection 11 with the reset counter 3. This connection 11 is provided to selectively transmit a refresh command to the reset counter 3 in the cases described previously.

[0053] The refresh command transmitted to the reset counter 3 can take on several forms within the scope of the invention. Provision may be made for this command

to be in the form of a pulse which restarts the countdown of the reset counter 3, whose duration is preset in the watchdog timer 1. Provision may also be made for the reset counter 3 to be programmable and for the refresh command given by the microcontroller 2 to define the duration of its countdown. The refresh command may therefore set a status of bits B0 to B5 to define the duration of the reset countdown. In the example shown in Figure 1, there are 64 different time period values to conduct reprogramming. In particular, provision may be made to use a reset counter 3 in the form of a register, in which the microcontroller 2 can only write on timeout of the refresh countdown or outside the time period of the refresh countdown.

[0054] The invention may in particular be applied using an external 8 MHz clock signal, a 1/50,000 frequency divider, a reset register with 7 programmable counting bits, and a refresh counter with 8 counting bits decremented directly by the external clock signal. The value initially recorded in the register may for example be programmed between FFh and C0h. In this case, the period of the reset countdown lies between 6.250 and 400 ms.

[0055] It is also preferably provided that the duration of the refresh countdown is less than the time period separating two decrements of the reset countdown. Provision may also be made in the method of the watchdog timer operation that this timer is disabled after it has generated a reset command. The user can therefore choose whether the application programmed by the microcontroller 2 is to have recourse to the watchdog timer 1 or not. This possibility is made possible in particular by the arrangement comprising the activation input 13 described above. Provision may also be made so that the watchdog timer 1

can only be disabled after it has generated a reset command for the microcontroller 2, to avoid erroneous deactivation by the microcontroller. Therefore, monitoring of the microcontroller 2 cannot be accidentally interrupted.

[0056] Provision may also be made for the microcontroller 2 to control its own resetting by software. For example, it may be provided that the microcontroller 2 commands an adequate write entry in bit B6 of the register so as to command its resetting. This possibility may be used in particular when the microcontroller 2 has itself detected a major error.

[0057] Although an independent watchdog timer 1 has been described above, the invention also applies to an integrated watchdog timer with other functions. For example, the watchdog timer 1 may be integrated in the microcontroller 2.